NitroGen: Rapid Development of Mobile Applications

Aharon Abadi[1], Yael Dubinsky[1], Andrei Kirshin[1], Yossi Mesika[1], Idan Ben-Harrush[1], Uzy Hadad[2]

[1] IBM Research – Haifa
[2] The Academic College of Tel Aviv-Yaffo
{aharona,dubinsky,kirshin,mesika,idanb}@il.ibm.com, uzy.hadad@gmail.com

Abstract

Constructing a mobile application is expensive and time consuming. In this paper, we present NitroGen which is a platform independent tool that provides a consumable integrated set of capabilities to construct mobile solutions aiming at reducing development and maintenance costs. NitroGen is a visual, mostly codeless, cloud-based platform to construct mobile applications. It can easily connect to back-end services thus enable fast and facile development in enterprises. Evaluating NitroGen, we found among others, that participants learned it fast and found it simple and suitable for mobile applications development.

Categories and Subject Descriptors D.2.2 [**Design Tools and Techniques**]: *User interfaces;* D.2.6 [**Programming Environments**]: *Graphical environments*

Keywords Mobile development, rapid application construction.

1. Presenter

Aharon Abadi's main research focus is on mobile software engineering. He belongs to the research group developing Nitrogen. He performed the evaluation of Nitrogen. His current research focuses on analysis and optimizations of Nitrogen applications. He joined IBM in 2007. His work has been included in IBM products. Aharon was awarded the IBM Research Division award (RDA) and the IBM Outstanding Innovation Award (OIA). He is the main author of more than 15 patents. He is the main organizer and the PC chair of the DeMobile 2013 workshop at FSE and MobileDeLi 2013 workshop at SPLASH.

2. Introduction

Building a mobile application is complex and expensive. Based on a study performed by Propelics [1] on the average of deployed mobile applications, there is an estimate of 1 week of effort per screen for a simple baseline, i.e for a simple 8-screen application the baseline is 8 weeks of effort, application visualization and prototype (2 weeks), development, unit testing, (4 weeks), quality assurance, user acceptance testing (2 weeks).

Developing an application involves obsessing over vast amounts of technical details due to today's fragmented and proprietary mobile market [2]. The proprietary programming languages, tools, models, and device variability cause a significant increase in development and maintenance costs. The development process becomes complicated for enterprises as there are critical decision points along the way that need broad technical perspective, as they significantly impact the developed application.

Finding an easy way to build client-side service-oriented applications is still a challenge. The existing state-of-the-art requires a deep understanding of technologies and internals, and the mobile aspect is adding a new set of challenges such as security, connectivity, and state synchronization. There are known existing development environments and frameworks for mobile, among them Xcode, ADT, Titanium, PhoneGap. These environments are for developers and require deep knowledge of the targeted technology. Further, these are no cloud-based and code-less environment. Most important, these environments do not provide easy access to back-end services. We suggest that there is a need to define new models that focus on the goal of the application itself, provide interface for non-developers and enable facile connectivity with the enterprise services.

Based on previous work [3], we present the NitroGen tool (see NitroGen screencast at https://ibm.biz/BdxLnu). NitroGen is a visual, mostly codeless (drag and drop), cloud based environment to construct mobile applications. The tool uses enterprise managed interfaces and provides a consumable, high-level integrated approach to building mobile applications that are not specific to any single platform or device. NitroGen allows solutions to be quickly constructed from customizable templates, instead of developing them from scratch. This solution accelerates mobile application construction, lowers the costs, and simplifies the development process by eliminating the need for heroic efforts or deep technical skills.

3. NitroGen Architecture

IBM Mobile Foundation® delivers a range of application development, connectivity and management capabilities that support a wide variety of mobile devices and mobile app types. However, a great deal of mobile application development skills is still required to construct a mobile application. These skills include, but not limited to: programming languages (Objective C, Java, C#, HTML, JavaScript, CSS), development environments (XCode, Eclipse, ADT, Visual Studio), platform specific APIs, frameworks (Dojo, jQuery, Backbone.js), communication with back-end, mobile specific user experience. NitroGen aims to eliminate technical skills that are required to develop mobile applications and extends IBM Mobile Foundation capabilities. It aims at making it easier for enterprise employees to create line of business applications that use and create data within their enterprise.

NitroGen is composed of design time components and runtime components. With respect to design time components, IBM WebSphere Cast Iron® is a graphical tool that enables users to integrate cloud and on-premise applications. The results are exposed to mobile applications in a form of services. Our contribution is based on IBM Forms Experience Builder®. With the help of this tool users of all skill levels can rapidly build and deploy data intensive web applications. NitroGen extends the tool for designing mobile applications. Services defined in Cast Iron are dynamically discovered and bound to the application data fields. NitroGen allows the user to

SPLASH '13, October 26–31, 2013, Indianapolis, Indiana, USA.
ACM 978-1-4503-1995-9/13/10.
http://dx.doi.org/10.1145/2508075.2514571

quickly preview the application in browser exactly as it will look like on a real device (various platforms are available). Then NitroGen generates the application code and publishes it to the IBM Worklight® for validation and testing on actual mobile devices. As for runtime components, tested applications are hosted and distributed using IBM Worklight. Applications securely communicate with Cast Iron via Worklight server. Cast Iron in its turn brings together the data from various sources.

4. Application Abstraction

To deal with the complexity associated with mobile application development, we have defined an abstract application model that represents a mobile application. The user designs the application with the help of a simple and intuitive web based Integrated Development Environment (IDE) never seeing any of the underlying models. Our runtime then interprets the application model providing the best user experience on different mobile platforms and form factors. While the created application is a true Model, View, Controller (MVC) type of application, the user of the IDE is unaware of this. She simply works with the user interface (UI) and connects the backend data to the UI. The user is unaware of how the application connects to the backend data as the IDE uses meta-data about the various backend data sources to generate the needed Worklight artifacts for this.

An application is a set of forms (screens), each having a sequence of fields that show application data. Fields are of different kinds, such as text, telephone, date, and image. Fields can be either read-only to present data or editable to collect data. Table is a special kind of field for showing repetitive data. Buttons and table rows are used for navigation between forms. Under the covers these UI artifacts are bound to a data model that is extracted from the UI designed by the user.

Services transfer data between the application and back-end. For example, the *Pre-population* service is automatically invoked when a form opens to fetch the needed data; the *Save* service is used to save form's data and its existence causes automatically add a save button to the UI in runtime. The NitroGen IDE's server has an extendable method for discovering data sources available to the user when creating the application. This process provides information such as connectivity and data fields available enabling an easy way to connect to the needed backend data sources.

The integration with the backend services in NitroGen is relevant in two stages of the application lifetime: design and runtime. At design time the application developer is binding the services to the application. The Cast Iron as a backend system provides SOAP based Application Program Interface (API) to query for available services and description of each service in a Web Service Definition Language (WSDL) format. The result of this discovery is a set of structured service description documents that contains the relevant information needed for the service binding. NitroGen platform use these service descriptors at design time to create the mapping dialog where developer is able to select the service and map fields to and from the service.

The same service descriptor is being used for generating the required adapters for the invocation of the service at runtime. Each adapter is configured to communicate with a specific backend system and acts as the mediator between the mobile application and the backend service. The main advantage of using an adapter is to take out the complexity of the integration from the client side to the server side. The use of an adapter is also crucial for overcoming security issues where the mobile application is not allowed to directly communicate with a backend service.

5. Evaluating NitroGen

We studied the functionality and usability of NitroGen with CS-major students in their third year of graduation who take the 'Developing Web Applications' course in The Academic College of Tel Aviv-Yaffo. In the first evaluation stage, participants were asked to fill in a pre-experiment questionnaire (10 minutes); In the second stage, they learned NitroGen while using the tool according to written tasks (90 minutes); and in the third stage they reflected on and shared their experience (10 minutes). In what follows, we present some of the findings.

Learning about their skills level using the pre-experiment questionnaire (36 participants' answers), we found that participants are knowledgeable with tools and techniques for developing web applications still do not have experience with development of mobile applications. In the second half of this questionnaire, we introduced a scenario to be developed as a mobile application and we asked participants to estimate the amount of time they need for the development of such a mobile application. The answers varied from 2 days to 3 months. We further asked about the most problematic issue in this application development. 11 participants indicated that developing the user interface is the most problematic issue, and 10 participants mentioned different features of the application. Only few (3 participants) refer to the database, to the suitability to different platforms (2 participants), or suggested that there are no problems (3 participants).

In the second stage, we presented the basic concepts of NitroGen to the participants, divided the class to groups (pairs and triplets), and provided a movie on developing using NitroGen and 9-tasks exercise that led step by step the development of a mobile application using NitroGen. We checked NitroGen server for the completion status of the different groups. 5 groups completed the exercise. 9 groups completed till a certain point among them 4 groups left at the early tasks and 5 groups stopped in the advanced part. It took 46 minutes for the first group to complete the exercise. After 75 minutes 5 groups completed the exercise.

In the last stage we asked participants to reflect on their development activity filling in a post-experiment questionnaire using close and open questions. Only 15 participants filled in the questionnaire, all of them are part of groups that completed the task. We found that participants like NitroGen features and found it not difficult to work with. In the last part of the questionnaire, we iterate the question on the estimation to develop the scenario. Answers ranged between 1 hour and 1 week. This is a significant change in how participants perceived of mobile application development. Since the scenario is relatively comprehensive and a small-scale application was developed in less than one hour, participants changed the way they estimated such a development.

We noticed that NitroGen is intuitive and easy to learn, and that no significant barrier was reported.

References

[1] Propelics, How to Size a Mobile App Development Effort, 2013 [online] http://www.slideshare.net/propelics/how-to-size-a-mobile-app-development-effort.

[2] Gavalas,D.,Economou,D.,Development platforms for mobile applications: status and trends. IEEE Software 28(1),2011.

[3] Shachor, G., Rubin, Y., Guy, N., Dubinsky, Y., Barnea, M., Kallner, S., Landau, A. What You See And Do Is What You Get: A Human-Centric Design Approach to Human-Centric Process, Business Process Design (BPD) in BPM, 2010.